

MICROPROCESSOR

Publication number: JP2042534 (A)

Publication date: 1990-02-13

Inventor(s): KISHIGAMI HIDEYA; MIYATA MISAO; OKAMOTO MITSUMASA

Applicant(s): TOKYO SHIBAURA ELECTRIC CO

Classification:

- international: G06F9/22; G06F9/38; G06F9/22; G06F9/38; (IPC-1-7): G06F9/22; G06F9/38

- European:

Application number: JP19890083243 19890331

Priority number(s): JP19890083243 19890331; JP19880078207 19880401

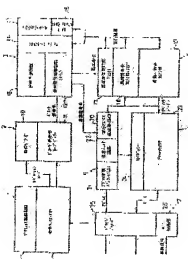
Also published as:

JP7120283 (B)

JP2085016 (C)

Abstract of JP 2042534 (A)

PURPOSE: To prevent the deterioration of the working efficiency at a prescribed stage of a pipeline by executing the instructions of the 1st and 2nd types in parallel with each other and independently of each other by a pipeline system. **CONSTITUTION:** The information equivalent to a high function instruction and a basic instruction is read out of an instruction decoding unit 2 and set at an instruction control circuit 12. The high function instruction is generated from the unit 2, and an execution address is calculated by an execution address generating part 20. An address converting buffer 21 performs the conversion of addresses, and the converted address is carried out by a high function instruction executing part 18. The result of execution of the part 18 is written into a future file 14. At the same time, the basic instruction is generated from the circuit 12 and executed by a basic instruction executing part 17. The result of this execution is written into the file 14. In such a way, the instructions are carried out in parallel with each other. As a result, the disturbance of a pipeline is suppressed and the performance of the pipeline is extremely improved.



Family list

2 application(s) for: JP2042534 (A)

1 MICROPROCESSORInventor: KISHIGAMI HIDEYA ; MIYATA MISAO Applicant: TOKYO SHIBAURA ELECTRIC CO
(+1)

EC: IPC: G06F9/22; G06F9/38; G06F9/22; (+3)

Publication info: JP2042534 (A) — 1990-02-13

JP7120283 (B) — 1995-12-20

JP2085016 (C) — 1996-08-23

2 MicroprocessorInventor: KISHIGAMI HIDECHIKA [JP] ; MIYATA Applicant: TOKYO SHIBAURA ELECTRIC CO [JP]
MISAO [JP]

EC: G06F9/38E; G06F9/38H2; (+1) IPC: G06F9/38; G06F9/38; (IPC1-7): G06F9/38

Publication info: US5155817 (A) — 1992-10-13

Data supplied from the esp@cenet database — Worldwide

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑩ 公開特許公報(A) 平2-42534

⑫ Int. Cl.³

識別符号

庁内整理番号

⑬ 公開 平成2年(1990)2月13日

G 06 F 9/38
9/22

3 7 0 B
3 5 0 E

7361-5B
7361-5B

審査請求 有 請求項の数 7 (全22頁)

⑭ 発明の名称 マイクロプロセッサ

⑮ 特 願 平1-83243

⑯ 出 願 平1(1989)3月31日

優先権主張 ⑰ 昭63(1988)4月1日 ⑱ 日本(JP) ⑲ 特願 昭63-78207

⑳ 発 明 者 岸 上 秀 哉 神奈川県川崎市幸区堀川町580番1号 株式会社東芝半導体システム技術センター内

㉑ 発 明 者 宮 田 雄 神奈川県川崎市幸区堀川町580番1号 株式会社東芝半導体システム技術センター内

㉒ 発 明 者 岡 本 光 正 神奈川県川崎市幸区堀川町580番1号 株式会社東芝半導体システム技術センター内

㉓ 出 願 人 株 式 会 社 東 芝 神奈川県川崎市幸区堀川町72番地

㉔ 代 理 人 弁理士 三好 秀和 外1名

明 細 書

1. 発明の名称

マイクロプロセッサ

2. 特許請求の範囲

(1) デコードされた命令のうち同一の処理過程を経て実行処理される第1の種類の命令をマイクロプログラム制御により実行処理する第1の実行処理手段と、

前記第1の種類の命令と処理過程が異なる第2の種類の命令をハードワイヤード制御により実行処理する第2の実行処理手段と、

デコードされた命令をプログラムシーケンスの順序で実行して、実行した命令を前記第1の実行処理手段および前記第2の実行処理手段のどちらで実行処理するかを選択決定し、前記第1の実行処理手段と前記第2の実行処理手段を独立にしかも並行して動作させる制御手段と

を有することを特徴とするマイクロプロセッサ。

(2) 前記第1の実行処理手段あるいは前記第2の実行処理手段により命令の実行が終了すると、前

記第1の実行処理手段あるいは前記第2の実行処理手段よりの実行結果を直ちに書込むための第1の情報保持手段と、

前記制御手段のプログラムシーケンス順序に従って前記第1の実行処理手段および前記第2の実行処理手段よりの実行結果を順序正しく書込むための第2の情報保持手段と

をさらに有することを特徴とする請求項(1)記載のマイクロプロセッサ。

(3) 前記制御手段により発行された命令に関する情報及びプログラムシーケンスにおける命令の実行/終了状態に関する情報を保持し、前記制御手段のプログラムシーケンス順序に従って順次正しく前記第2の情報保持手段を更新するための第3の情報保持手段と

をさらに有することを特徴とする請求項(2)記載のマイクロプロセッサ。

(4) 前記第1の種類の命令が、メモリアクセスを有する処理の複雑な高機能命令であり、前記第2の種類の命令が、メモリアクセスを持たな

い基本命令であることを特徴とする請求項(1)に記載のマイクロプロセッサ。

(5) プログラムシーケンスでは先の前記第一の命令実行処理手段で実行処理される第1の種類の命令より、プログラムシーケンスでは後の前記第二の命令実行処理手段で実行処理される第2の種類の命令の方が先に実行処理を終了することを特徴とする請求項(1)に記載のマイクロプロセッサ。

(6) 前記第一の情報保持手段および前記第二の情報保持手段の役割が固定式的ではなく、前記第一の情報保持手段の記憶要素および前記第二の情報保持手段の記憶要素共に、ダイナミックにその役割が切替わることを特徴とする請求項(1)に記載のマイクロプロセッサ。

(7) 前記第一の命令実行処理手段で実行処理される命令および、その命令にプログラムシーケンス上で使われる前記第二の命令実行処理手段で実行処理される命令(列)に対して、同一の認識番号を付加する手段と、

前記第一の命令実行処理手段で実行処理中の命

令の認識番号を保持する第四の情報保持手段と、前記第一の情報保持手段の記憶要素および前記第二の情報保持手段の記憶要素の役割等の情報を保持する第五の情報保持手段と、

前記第一の情報保持手段および前記第二の情報保持手段の読み出し/書き込み信号および前記第五の情報保持手段の更新を行う更新手段と

をさらに有することを特徴とする請求項(1)に記載のマイクロプロセッサ。

3. 発明の詳細な説明

【発明の目的】

（産業上の利用分野）

この発明は、命令をパイプライン方式により実行処理するマイクロプロセッサに関し、特に、パイプラインの乱れを抑制して、性能を大幅に向上させることができるマイクロプロセッサに関する。

（従来の技術）

近年、マイクロプロセッサにおいては、命令をパイプライン方式より実行処理して、性能の向

上を図っている。このパイプライン方式における一般的なステージの構成は、例えば「命令フェッチ→命令デコード→実効アドレス計算→アドレス変換→オペランドリード（読出し）→命令実行→オペランドライト（書込み）」となる（文献「32ビット・マイクロプロセッサの全書—企業・戦線・技術・市場動向」日経マグロウヒル社、P. 137～139）。

このようなパイプライン構成において、メモリオペランドを有する高機能命令(1)は、実効アドレスの計算及び実効アドレスから物理アドレスに変換を行うアドレス変換のステージでの処理が必要となる。これに対して、メモリオペランドのない基本命令(1R)では、上記2つのステージでの処理は不要となる。

したがって、例えば命令のシーケンスが、1→1R→1→1R→1→1→1R→1Rのような場合には、パイプラインの“流れ”は、第12図に示ようになる。なお、各ステージの処理は1サイクルで終了するものとし、命令(1)のオペラン

ドライトをレジスタとして、実行ステージで完了するものとする。また、第12図において、X印はステージの動作が休止状態であることを示している。

第12図から明らかなように、実効アドレス計算のステージ(OAG)は、4サイクル目と6サイクル目において休止状態であり、アドレス変換のステージ(MAM)は、5サイクル目と7サイクル目において休止状態となっている。

このことから、実効アドレス計算及びアドレス変換の各ステージでの稼働率は、50(%)となる。

一方、複雑な高機能命令セットを有するCISC(Complex Instruction Set Computer)型のマイクロプロセッサの場合には、実行のステージでの処理に数サイクルを必要とする複雑な高機能命令(1c)がある。

このようなマイクロプロセッサにおいて、例えば命令シーケンスが、1c→1R→1R→1R→1R→1Rのような場合は、パイプラインの流れが第1

3図に示すようになる。なお、第13図において、命令1cは、その実行ステージでの処理に4サイクルかかるものとし、X印は第12図と同様とする。

このような場合には、命令1cの実行に4サイクルかかるために、第13図から明らかなように、所謂「パイプラインの乱れ」が生じる。これにより、第13図に示した例では、すべての命令の実行が第13図の斜線で示した理想的なパイプラインのずれの中で終了せず、3サイクル分(12サイクル目～14サイクル目)だけ処理が長くなっている。

また、高機能命令1cの実行に4サイクルかかるため、実行アドレス計算(OAG)、アドレス変換(MMU)及びオペランドリード(OP)の各ステージにおいて、休止状態が存在することになる。

(発明が解決しようとする課題)

パイプライン処理を行うマイクロプロセッサにおいて、メモリオペランドを有する高機能命令

(1a)とメモリオペランドのない基本命令(1b)がそれぞれ交互に実行された場合には、乱れは生じない。しかし、第12図に示したように、実行アドレス計算及びアドレス変換のステージでの遅延率が低下するという問題が生じる。

また、実行ステージでの処理に数サイクルを必要とする複雑な高機能命令(1c)が実行される場合には、パイプラインの乱れに乱れが生じる。これにより、性能が低下するという問題があった。さらに、このような場合にも、所定のステージでの遅延率が低下することになる。

そこで、この発明は、上記問題に鑑みてなされたものであり、その目的とするところは、ステージの遅延率の低下を防止するとともに、パイプラインの乱れを抑制して、性能を大幅に向上させることのできるマイクロプロセッサを提供することである。

【発明の構成】

(課題を解決するための手段)

上記目的を達成するために、この発明に従う

マイクロプロセッサは、デコードされた命令のうち同一の処理過程を経て実行処理されるメモリオペランドを有する処理の複雑な高機能命令をマイクロプログラム制御により実行処理する第1の実行処理手段と、前記第1の種類の命令と処理内容が異なるメモリオペランドを持たない基本命令をハードワイヤード制御により実行処理する第2の実行処理手段と、デコードされた命令をプログラムシーケンスの順序で発行して、発行した命令を前記第1の実行処理手段および前記第2の実行処理手段のどちらで実行処理するかを選択決定し、前記第1の実行処理手段と前記第2の実行処理手段を独立にしかも並行して動作させる制御手段とを有している。

そして、この発明に従うマイクロプロセッサは、さらに、前記第1の実行処理手段あるいは前記第2の実行処理手段により命令の実行が終了すると、前記第1の実行処理手段あるいは前記第2の実行処理手段よりの実行結果を直ちに蓄積するための第1の情報保持手段と、前記制御手段のプログラム

シーケンス順序に従って前記第1の実行処理手段および前記第2の実行処理手段よりの実行結果を順序正しく蓄積するための第2の情報保持手段と、前記制御手段により発行された命令に関する情報及びプログラムシーケンスにおける命令の実行/終了状態に関する情報を保持し、前記制御手段のプログラムシーケンス順序に従って順序正しく前記第2の情報保持手段を更新するための第3の情報保持手段とを有している。

(作用)

上記構成のマイクロプロセッサによれば、高機能命令と基本命令を、それぞれ独立して実行処理するようにして、高機能命令と基本命令が並行してあるいは同時に実行処理されることを可能にしている。

また、制御手段によって発行された命令の実行を、第1の情報保持手段の更新にしたがって開始するようにしている。さらに、メインルーチンからはずれたサブルーチンが実行された後、プログラムシーケンスがメインルーチンに戻った時に、

実行開始の命令を第2の情報保持手段の保持内容にしたがって決定するようにして、命令を再実行できるようにしている。

また、発行された命令に関する情報及び命令の実行/終了状態に関する情報にしたがって、第2の情報保持手段の保持内容がプログラムシーケンス順に順序正しく更新されるようにしている。

(実施例)

以下図面を用いてこの発明の実施例を説明する。

第1図は、この発明を実施したマイクロプロセッサの内部全体の構成を示すブロック図である。

このマイクロプロセッサは、主記憶からの命令データのフェッチを行う命令フェッチユニット (IFU) 1と、上記命令フェッチユニット1よりの命令データの解読を行うためのデコードユニット (DCU) 2と、上記デコードユニット2から送られてきた命令情報をその種類すなわち、メモリオペランドを持たない基本命令およびメモリオペランドを持つ基本命令あるいは処理の種類な

高優先命令に従って発行するための命令発行ユニット (IU) 3と、命令の実行を上記種類に従ってハードワイヤード制御またはマイクロプログラム制御で行うための命令実行ユニット (EXU) 4と、メモリオペランドのアドレスを生成するためのメモリ管理ユニット (MMU) 5と、オペランドデータを管理するためのキャッシュ制御ユニット (CCU) 6と、上記マイクロプロセッサと外部とのデータ入出力を制御するための入出力部 (I/O) 7とを有している。

上記命令フェッチユニット (IFU) 1は、主記憶上の命令データ群の一部のコピーを保持する命令キャッシュ・メモリ (Instruction Cache) 8や命令キャッシュ・メモリ8への主記憶からの命令データのフェッチ等の制御を行うプリフェッチ制御回路 (Prefetcher) 9等から構成されるもので、従来と同様のものである。

上記デコードユニット (DCU) 2は、命令コードの解読を行う命令デコード (Decoder) 10やデコードした結果の命令情報を複数回、一時的

に保持するデコード済命令ループバッファ (Decoded Instruction Loop Buffer) 11等から構成される。本実施例ではデコードした命令情報をデコード済命令ループバッファ11から一度 (1サイクル) に2命令分読み出し、命令発行ユニット (IU) 3に転送である構成となっている。

ただし本発明には、デコード済命令ループバッファ11や一度に2命令分読み出す機能は必ずしも必要ではない。

上記命令発行ユニット (IU) 3は、上記デコードユニット2から送られてきた命令情報と、上記種類にしたがって、命令実行ユニット (EXU) 4ないしメモリ管理ユニット (MMU) 5に対して発行する命令発行制御回路 (Instruction Issue Logic) 12や汎用レジスタ値を保持するカレントファイル (Current File) 13、フューチャファイル (Future File) 14、およびリオーダーバッファ (Reorder Buffer) 15等から構成される。

上記命令発行制御回路 (IIL) 12は通常のパイプライン処理を行なうマイクロプロセッサがもつパイプライン制御回路の機能 (ハザードの検出などを行ない、各パイプライン・ステージの状態制御を行なう) のほかに、上記送られてきた命令情報が、メモリオペランドを持たない基本命令であるかメモリオペランドを持つ基本命令あるいは処理の種類な高優先命令であるかを選択決定し、後述する複数の命令実行部において上記各命令が並行して実行される様に制御する機能、後述する複数の命令実行部でプログラムシーケンス順とは異って終了する命令実行結果の情報をプログラムシーケンス順に異なすためリオーダーバッファ15の制御 (情報設定/解除) を行なう機能を有する。上記カレントファイル13はプログラムシーケンス順に従って更新されるが、フューチャファイル14はプログラムシーケンス順とは無関係に後述する命令実行ユニット (EXU) 4で実行終了後、その実行結果によってただちに更新される。上記リオーダーバッファ15は命令実行ユニット (E

XU) 4の複数の命令実行部でプログラムシーケンス順とは異って終了する命令実行結果の情報を一時保持し、プログラムシーケンス順にカレントファイル13を更新するためのバッファである。

すなわち、上記基本命令と高機能命令とは実行に要するサイクルが異なり、ここでは、実行に要するサイクルが異なる命令をそれぞれに対応した命令実行部で実行するようにしているため、プログラムシーケンスの順序で実行される命令は、モ実行が必ずしもプログラムのシーケンスの順序にしがって終了するとは必ずしも、順序が逆転することがある。

したがって、リオーダーバッファ15は、プログラムシーケンスの順序でカレントファイル13の中のレジスタの内容を更新して、上記の逆転した順序をプログラムシーケンスの順序に戻すようにしている。すなわち、Out of order で終了した命令をReorderする働きをする。

これにより、新込み等のメインルーチンからはずれたプログラムが実行された場合には、カレン

トファイル13の内容を参照することにより、命令を再実行することが可能となる。

また、上記命令発行ユニット(11U)3は、分岐命令の高速実行を行うための分岐予測回路(Branch Prediction Logic)16等も有する。

上記命令発行制御回路12、カレントファイル13、フューチャファイル14、およびリオーダーバッファ15は本発明の目的を達成するために必要な構成要件である。ただし、リオーダーバッファ15を用いないで本発明の目的を達成する方法もあり、それについては他の実施例ということであらう。また上記カレントファイル13とフューチャファイル14は物理的には必ずしも別のものでなくとも良く、ひとつのレジスタファイルを2つの部分に分けた場合の一方と他方でも良く、それについても他の実施例ということであらう。

上記命令実行ユニット(EXU)4は、命令の実行をハードワイヤード制御またはマイクロプログラム制御で並行して行なうユニットである。この

実施例では、メモリアペラントを持たない基本命令(比較・転送命令・算術・論理演算命令など)をハードワイヤード制御で行なう基本命令実行部(Simple Execution processor)17、メモリアペラントを持つ基本命令や処理の複雑な高機能命令をマイクロプログラム制御で実行する命令実行部(Integer Execution Processor)18、および浮動小数点演算命令を実行する浮動小数点実行部(Floating Execution Processor)19の3つの実行部から構成される。

なお本発明は、命令の種類に対応した複数の命令実行部を持つことが特徴であり、必ずしも3つの実行部から構成されなくとも良い。また本発明の变形として、メモリアペラントを持たない基本命令の実行部とオペラントの実効アドレス計算を行なう部分を共通化した構成も考えられる。

上記メモリ管理ユニット(MMU)5は、メモリアペラントの実効アドレスを生成する実効アドレス生成部(Operand Address Generator)20、実効アドレス(論理アドレス)を物理アド

レスに変換するアドレス変換バッファ(Translation Lookaside Buffer)21、メモリ保護のチェックを行う保護チェック回路(Protection Logic)22等から構成されるもので、従来と同様のものである。

上記キャッシュ制御ユニット(CCU)6は、主記憶上のオペラント群の一部のコピーを保持するデータキャッシュ・メモリ(Data Cache)23や書き込みオペラントデータを一時保持するストア・バッファ(Store Buffer)24等から構成されるもので、従来と同様のものである。

上記入出力部(I/O)7は、マイクロプロセッサと外部とのデータ入出力を制御する部分でドライバ/レシーバ(Driver/Receiver)25やバス制御部(Bus Control)26等から構成されるもので、従来と同様のものである。

第2図は、第1図に示したマイクロプロセッサの内部ブロックにおいて本発明に係る関連する制御ブロックを示したものである。

第2図において、バスは2重線で示し、データ

線は直線で示しており、制御線は省略している。

そして、第2図における各ブロックの内部をさらに詳細に示すと第3図の如くなる。

第3図において、前記命令実行制御回路(11L)1・2は、パイプラインの各ステージで実行中の命令に関する情報を保持するパイプライン・レジスタ(OACR30、MMUR31、CCUR32、IEPR33およびSEPR34)と、それらの情報を元にパイプラインの流れを制御するコントロール回路(Control)35から構成される。パイプラインの流れについては第7図、第8図を参照して後述する。コントロール回路35はまだりオーダバッファR15の制御(データの登録・削除等)も行う。

本実施例では前記命令実行制御回路12は、1サイクルで2命令分の情報をデコードユニット2のデコード済命令ループバッファ(DILB)1から受けることができる。(ただしそのうち1つはメモリ・オペランドを持たない基本命令。)

SEPR34は現在基本命令実行部17で実行

中の命令に関する情報を保持するレジスタである。

OAGR30は現在OAG20で実効アドレス計算中の命令に関する情報を保持するレジスタである。

MMUR31は現在MMU5でアドレス変換中の命令に関する情報を保持するレジスタである。CCUR32は現在CCU6でメモリ・アクセス(オペランドリード)中の命令に関する情報を保持するレジスタである。

IEPR33は現在IEP18で実行中の命令に関する情報を保持するレジスタである。

なおオペランド・ライトに関する情報は、CCU6のストアバッファ24に保持されるため11L12にはオペランド・ライトに関する情報を保持するレジスタは存在しない。

11L12の詳細ブロックは第4図参照。

基本命令実行部(SEP)17はメモリ・オペランドを持たない基本命令をハードワイヤ制御で実行する為の演算器(Address)36を持つブロックである。演算器36は11L12中のSEP

R34によって直接制御される。

高機能命令実行部(IEP)18は、高機能命令をマイクロプログラム制御で実行するための演算器(ALU37、Barrel Shifter38、Multiplier39)および、マイクロプログラムを保持するμROM40およびシーケンサから構成される。RAL41はμROM40のアドレスを保持するためのレジスタ、MIR42はマイクロ命令を保持するためのレジスタ、ErrAddr43はエラー発生時のμROM40のアドレスを保持するためのレジスタである。またSEL44はRAL41、ErrAddr43および11L12のCCUR32のオペランドR8に保持されている値(次にIEP18で実行する命令の先頭マイクロ命令のアドレス)のうちの一つを選択するためのセレクトである。

実行アドレス生成部(OAG)20は、メモリ・オペランドの実効アドレスを算出するための加算器(Address Generator)47から構成される。

メモリ管理ユニット(MMU)5は、論理アド

レス(実効アドレス)を物理アドレスに変換するためのアドレス対を保持するアドレス変換バッファ(Translation Lookaside Buffer: TLB)21およびメモリ・アクセス権をチェックするためのアクセス権チェック回路(Protection Logic)22から構成される。

キャッシュ制御ユニット(CCU)6は、メイン・メモリのデータの一部分のコピーを保持するデータ・キャッシュ(Cache)23およびライト・データの情報を一時的に保持するストア・バッファ(Store Buffer)24より構成される。データ・キャッシュ23はデータを保持するデータ部(DATA)48とアドレスや属性を保持するタグ部(TAG)49から成る。またストア・バッファ24はデータを保持するデータ部(DAT)50とアドレスを保持するアドレス部(ADDRESS)51より成る。IEP18より送られてきたライト・データはいったんストア・バッファ24に格納され、その後データ・キャッシュ23および主記憶に書き込まれる。

第4図は、第3図中の11112、RB15、CP13、FF14の部分の詳細図である。

DCU2のデコード済命令ループバッファ11から送られてきた命令の情報は、SEPR34またはOAGR30に格納される。SEPR34にはメモリ・オペランドをもたない基本命令の情報の格納することができる。一方OAGR30には全ての命令の情報を格納することができる。

SEPR34は次のフィールドから構成される。
OP60…基本命令の種類を示し(比較、転送、加算など)、SEPの演算器の機能を制御する。
R/161…ソース・オペランドがレジスタかイミディエイト・データかを区別する。
#src62…ソース・オペランドのレジスタ番号を指定する。
#dest63…デスティネーション・オペランドのレジスタ番号を指定する。
1ss64…イミディエイト・データ。
PC65…命令の先取アドレス
V66…有効ビット

OAGR30に格納された命令の情報は、命令パイプラインの各ステージでの処理が進むにつれて、OAGR30→MMUR31→CCUR32→IEPR33と転送される。

OAGR30→MMUR31では、OAG20で、Anode73、Areg74、Disp75の情報に基づいて実効アドレス(論理アドレス)の計算が行われる。

MMUR31→CCUR32では、MMU5で、論理アドレスが物理アドレスに変換される。またメモリ・アクセス権のチェックが行われる。CCUR32→IEPR33では、OPフィールド88で、#ROM40のアクセス(命令を実行する先頭マイクロ命令の読みだし)が行われる。

図中の制御回路(Control)35は、SEPR34、OAGR30、MMUR31、CCUR32、IEPR33に接続されている命令の情報および、以下の信号を入力してパイプラインの状態制御、ハザード検出、リオーダ・バッファ(RB)

OAGR30は次のフィールドから構成される。
OP67…命令の種類を示す。

R/M168…ソース・オペランドがレジスタかメモリかを区別する。

#src69…ソース・オペランドのレジスタ番号を指定する。

R/M70…デスティネーション・オペランドがレジスタかメモリかを区別する。

#dest71…デスティネーション・オペランドがレジスタかメモリかを区別する。

1ss72…イミディエイト・データ。

Anode73…メモリ・オペランドのアドレッシング・モードを指定する。

Areg74…メモリ・オペランドのアドレッシング・モードで使用するレジスタ番号を指定する。

Disp75…メモリ・オペランドのアドレッシング・モードで使用するディスプレースメント。

Ex.76…その他。

PC77…命令の先取アドレス。

V78…有効ビット。

15の制御信号を生成する回路である。制御回路35の詳細については第5図、第6図を参照して決する。

ストアバッファ・ビジー信号(Store Buffer Busy)102

μプログラム終了信号(μEND)103

キャッシュ・ミス信号(Cache miss)104

μ命令でのGRへのライト信号(μ→GR)105

カレント・ファイル(CF)13はプログラムシーケンス順に従って更新される汎用レジスタ値を保持するレジスタ・ファイルであり、フェューチャ・ファイル(FF)14はSEPR17/IEPR18での命令終了によりただちに更新される汎用レジスタ値を保持するレジスタ・ファイルである。

リオーダ・バッファ(RB)15は、SEPR17とIEPR18の2つの命令実行部でプログラムシーケンス順とは異なって終了する命令実行結果を一時的に保持し、プログラムシーケンス順にCF13を更新するためのバッファである。本実施例で

はRB15はBエントリであり、以下のフィールドから構成される。

State 106…エントリの有効/無効および実行中/実行終了を示す。

R/M107…命令のデスティネーションがレジスタかメモリかを示す。

#dest 108…デスティネーションがレジスタの場合のレジスタ番号を示す。

Result 109…命令の実行結果を保持する。

Flag 110…命令の実行結果のフラグを保持する。

Error 111…命令の実行結果でエラーがあった場合のエラー情報を示す。

PC112…命令の先頭アドレス。

RB15への情報の登録は、SEPR34に保持されている命令がSEPR34で実行されるタイミングないしは、OAGR30に保持されている命令がMMUR31に転送されるタイミングで行われる。図中のtail 113、head 114は各々RB15に登録された最も新しい命令情報を保

持するエントリ+1、および最も古い命令情報を保持するエントリをポイントするレジスタである。

RB15へは1サイクルでtail 113がポイントするエントリおよびtail+1がポイントするエントリに2命令分の情報を同時に登録できる。またRB15からは、head 114がポイントするエントリのState 106が実行終了状態であれば、そのエントリのResult 109、Flag 110に保持されている実行結果にしたがってCF13およびFlagレジスタ115の値が更新される。またError 111にエラー情報がある場合には、μプログラム、シーケンス制御部にエラー信号を発生し、エラー処理のμプログラム・ルーチンを起動する。RB15からのデータ読み出しは1サイクルで最大1命令分行うことができる。

RB15へ命令情報を登録した時にはtail 113は+1ないし+2カウントアップされる。またRB15のデータ読みだしが行われたときはhead 114は-1カウントダウンされる。

第5図は、第4図中の制御回路135の内部部

ロックをしめたものである。制御回路135は、パイプライン・レジスタのレジスタに関する情報を基にハザード・チェックを行う部分と、パイプライン・レジスタの有効信号、ハザード・チェック信号等を基にパイプラインの状態制御を行う状態制御回路(State Control Circuit)120から構成される。

図中のハザード F/F121は、16ビットのレジスタで、パイプライン・レジスタ(MMUR31、CCUR32、JEP33)の命令が汎用レジスタに結果を書き込むとき、対応するビットに1がセットされていて、この情報を基にハザード検出を行う。ハザード F/F121は、OAGR30のR/M70、#dest71をデコード124でデコードした結果でセットされ、JEP33のR/M97、#dest98をデコード127でデコードした結果でリセットされる。

SEPR34に保持されている命令がSEPR17で実行できる条件は、ソース/デスティネーションに使用するレジスタともに書き変わる可能

性がないときである。(すなわちハザード F/F121の対応するビットに1がたっていないとき)この条件の検出は、デコード122、デコード123でSEPR34のR/161、#src62および#dest63をデコードした結果とハザード F/F121値とを比較回路CMP1128、CMP2129で比較しその結果のOR出力信号(ハザード(SEP))133で行う。この条件が満足されるとハザード(SEP)133が0となり、満足されないときはハザード(SEP)133が1となる。

同様にしてOAGR30のAcode73、Areg74をデコード125でデコードした結果とハザード F/F121の値を比較回路CMP130で比較し、その出力信号(hazard(OAG))134が0のときOAG20で有効アドレスの計算が可能となる。

またCCUR32のR/M89、#src90をデコード126でデコードした結果とハザード F/F121の値とを比較回路CMP41

31で比較し、その出力信号(ハザード(CCU)
135が0のときソース・オペランド(レジス
タ)の読み出しが可能となる。

状態制御回路(State Control Circuit)
120は、上でのべた3つのハザード信号(haza
rd(SEP)133、hazard(OAG)134、
hazard(CCU)135)、パイプライン・レジ
スタの有効信号(V(IEP)101、V(CCU)
96、V(MMU)87、V(OAG)78)
やレジスタ/メモリ番号(R/M(IEP)97、
R/M1(MMU)89、R/M2(MMU)9
1)およびILL12外部からの信号(ストア
バッファビジー信号102、μ-VCR105、C
ache miss 104、μ-VCR105)を入力
して、パイプラインの状態制御を行う以下の信号
を出力する。

SEP-136 SEP PRに保持されている命令
がSEPで実行可能となるとき1になる。

OAG-MMU137 OAG PRに保持されてい
る命令が次サイクルでMMURに進めるとき1に

なる。

MMU-CCU138 MMURに保持されてい
る命令が次サイクルでCCURに進めるとき1に
なる。

CCU-IEP139 CCURに保持されてい
る命令が次サイクルでIEPRに進めるとき1に
なる。

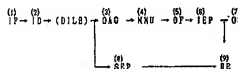
IEP-SB140 IEPRに保持されている
命令が次サイクルでストアバッファに情報を転送
するとき1になる。

第6図は、第5図中のステートコントロール回
路の具体的な回路例である。

次に、第1図を参照して上記、本発明に従うマ
イクロプロセッサのパイプライン処理動作の概略
について説明する。

すなわち、パイプライン処理動作の概略は以下
のようになる。

(以下余白)



(1) IF(命令フェッチ)ステージ

IFUにおいて、命令キャッシュメモリ8から
の命令のフェッチを行うステージ。

(2) ID(命令デコード)ステージ

DCUにおいて、命令デコード10で命令のデ
コードを行い、内部命令フォーマットに変換する。
なお内部命令フォーマットに変換された命令はデ
コード済みループバッファ11に格納される。
そして、内部命令フォーマットはメモリ・オペラ
ンドを持たない基本命令とメモリ・オペランドを
持つ基本命令ないしは高機能命令の2種類あり、
命令発行制御回路(ILL)12によってそれぞれ
発行される。

(3) OAG(オペランド実効アドレス算出) ステージ

OAG20のアドレス発生回路47で、命令発

行制御回路(ILL)12によって発行されたメ
モリオペランドを持つ命令のメモリ・オペランド、
実効アドレス(算出アドレス)を算出するステー
ジ。

(4) MMU(アドレス変換)ステージ

MMU5のアドレス変換バッファ21で、メモ
リ・オペランドの論理アドレスを物理アドレスに
変換するステージ。また保護チェック回路22で
メモリ保護のチェックも行われる。

(5) OF(オペランド・フェッチ)ステージ

CCU6のデータ・キャッシュメモリ23から
メモリ・オペランドを読み出すステージ。また、
レジスタオペランドの読み出しも行われる。

(6) IEP(命令実行)ステージ

EXU4の高機能命令実行部(IEP)18に
おいて、μプログラム制御でメモリオペランドを
持つ基本命令あるいは高機能命令を実行するス
テージ。

(7) OS(オペランド・ストア)ステージ

IEP18での実行結果をCCU6のストア・

バッファ24に書き込むステージ。ただしこのステージがあるのは、命令のディスティネーションがメモリの場合のみ。なお演算結果はストア・バッファ24を介して、データ・キャッシュメモリ6とマイクロプロセッサ外部の主記憶に、パイプライン処理とは非同期に書き込みが行われる。

(4) SEP (命令実行) ステージ

EXU4の基本命令実行部(SEP)17において、ハードワイヤード制御で命令発行制御回路(ILL)12によって発行された基本命令を実行するステージ。なおSEP17で実行される命令は、メモリ・オペランドを持たない基本命令のみ。

(5) RE (リオーダ) ステージ

IEP18およびSEP17よりの実行結果をリオーダバッファ(RB)15によりリオーダしてカレントファイル(CF)13に書き込むステージ。

以上のパイプライン処理のうち、(4) IEPを除く他のステージは、基本的には1サイクルでそ

の処理が終了する。ただしキャッシュ・ミス、TLBミスが生じたときには、(1) IF、(4) MMU、(5) OFのステージの処理も複数サイクル必要となる。また、ハザード(例えば、IEPステージの実行結果を実効アドレス算出に使用する等)が生じたときには、いわゆる「待ち」が生じて1サイクルで処理が終了しなくなる。

本発明の特徴は、複数の命令実行部を持ち命令の並列実行を可能とすることである。すなわち本実施例では、主に、SEPステージおよびRESステージが新たに加わった点が従来技術と比べて新しい。

次に、第7図および第8図を参照して、上記本発明の特徴的な処理動作をさらに詳細に説明する。

第7図および第8図は、本発明の実施例(すなわち、基本命令実行部SEP17がある場合)のパイプラインタイミング図をそれぞれ示し、第7図のタイミング例は、第12図に示した従来のタイミング例に対応し、第8図の例は、第13図の従来のタイミング例に対応する。

ただし第7図、第8図では簡潔のために、デコード済命令ループバッファ(DILB)11以降の部分のみを示し、命令はDILB11中に有るものと仮定する。

第7図は、命令シーケンスが

```

[al -> IR2 -> IR3 -> IR4 -> IR5 -> IR6

```

の場合のパイプライン・タイミング例-1である(I m)と基本命令(IR)は(DILB)から1サイクル同時に読み出されたものとする。またIm1とIm3はディスティネーションがレジスタ、Im5はディスティネーションがメモリとし、ハザードは生じないものとする。サイクル1ではIm1、IR2の2命令分の情報(DILB11から読み出され、ILL12のOAGR30およびSEP1734レジスタにセットされる。

命令Im1は、サイクル2で命令発行制御部12によって発行され、実行アドレス生成部(OAG)20によって実行アドレス算出が行われ、サイクル3でアドレス変換バッファ(TLB)21

によってアドレス変換が行われ、サイクル4でメモリ管理ユニット(MMU)5によってオペランドフェッチが行われ、サイクル5で高機能命令実行部(IEP)18によって実行され、サイクル6でディスティネーションがレジスタのためフューチャファイル(F F)14へその実行結果が書き込まれる(第7図のF Fの欄のIm1を参照)。

一方、これと並行して、基本命令IR2は、サイクル2で命令発行制御部12によって発行され、基本命令実行部(SEP)17によって実行され、サイクル3でフューチャファイル14へその実行結果が書き込まれる(第7図F Fの欄のIR2を参照)。

ここで、リオーダバッファ(RB)15への命令情報の登録は、高機能命令Imは、実行アドレス算出ステージで行われ、基本命令IRは、基本命令実行部17での実行ステージで行われるため、Im1およびIR2の情報は、図示する如く(サイクル3で登録される。第7図のRBの欄の命令の

上の“X”および“●”印は、命令が各々“実行中”および“実行終了”であることを示している。

一方、リオーダバッファ15からカレントファイル(CF)13への命令実行結果の書き込みは、リオーダバッファ(RB)15において命令情報に削除されたサイクルで行われる。従って、1m1の場合は、その命令情報がサイクル7でリオーダバッファ15から削除されているため、サイクル7で、その実行結果が、カレントファイル(CF)13へ書き込まれる。また、1R2の場合は、その命令情報が、サイクル8でリオーダバッファ15から削除されているため、サイクル8でその実行結果が、カレントファイル(CF)13へ書き込まれることとなる。

すなわち、フューチャファイル(FF)14は、命令実行後だけに更新(書き込み)されるため、プログラムシーケンス順とはなっていないが、カレントファイル(CF)13は、リオーダバッファ15から命令情報が削除されるタイミングで更新されるためプログラムシーケンス順に命令実

行結果がファイルされていく。

命令1m3、1R4、1m5、1R6の場合も、上述したと同様に処理されるものである。

第8図は、命令シーケンスが
1c1 → 1R2 → 1c2 → 1R3 → 1c3 → 1R4 → 1c5
の場合のバイブリン・タイミング例-2である。この例の場合も1c1の実行に時間がかかっているが、その間に1R2、1R3、1R4の実行は基本命令実行部(SEP)17で先に終了している。

すなわち、高機能命令1c1は、サイクル2〜4で、実行アドレス算出、アドレス変換、およびオペランドフェッチが行われ、サイクル5〜8で高機能命令実行部(IEP)18によって実行され、サイクル9でフューチャファイル(FF)14へその結果が書き込まれる。

一方、これと並行して、基本命令1R2は、サイクル2で基本命令実行部(SEP)17によって実行され、サイクル3でフューチャファイル(FF)14へその実行結果が書き込まれる。

ここで、リオーダバッファ(RB)15からカレントファイル(CF)13への命令実行結果の書き込みは、リオーダバッファ(RB)15において命令情報が削除されたサイクルで行われる。

従って、第7図に示した例と同様に、カレントファイル(CF)13には、プログラムシーケンス順に命令実行結果がファイルされるものである。

以上、第7図および第8図の例からわかるように、本発明では複数の命令実行部を持ち命令の並列実行することにより、従来例で生じていたパイプラインの乱れを抑え、また各パイプライン・ステージの稼働率の低下を抑えることができ、結果として大幅な性能向上を得ることができる。

また、通常の命令実行状態において、フューチャファイル(FF)14に保持されている汎用レジスタ値は、カレントファイル(CF)13に保持される汎用レジスタ値と異なっている。これはプログラム・シーケンス順では後のメモリ・オペランドを持たない基本命令が、プログラム・シーケンス順では前の高機能命令実行部(IEP)

18で実行されるメモリ・オペランドを持つ高機能命令より先に基本命令実行部(SEP)17で実行され、フューチャファイル(FF)14を更新するためである。ただし、高機能命令実行部(IEP)18で実行した命令でエラー(制込み)が発生した場合には、命令の再実行を保证するためにフューチャファイル(FF)14の値をカレントファイル(CF)13の値に戻さなければならぬ。このためにカウンタ119が用意されている。制込み処理プログラム・ルーチンではカウンタ119を利用してカレントファイル(CF)13の値をフューチャファイル(FF)14にコピーすることができる。

第9図に発明の実施例を適用したMFPUと周辺LSIから成るシステム構成例を示す。この例はVMEバス200につながる比較的簡単なシステムであり、以下のLSI、ICから構成される。
MFU201

ICT202—制込みコントローラ
CG203—クロック・ジェネレータ

メモリ…SRAM (0ウェイト 32Kバイト)
 204
 EPROM (0ウェイト 32Kバイト)
 205
 DRAM (3ウェイト 4Mバイト)
 206
 通信インタフェース…セントロニクス 1チャネル
 207
 RS232C 2チャネル
 208

その他…T/R トランシーバ/レシーバ209
 Buf バッファ210, 211
 Decode アドレス・デコーダ212

本発明を使用したMPUを使用したシステム構成は、従来のMPUを使用したシステム構成と何ら異なることはない。すなわち本発明を使用したMPUを使用することによりシステム・レベルで必要な付加回路は無く、高性能なシステムを構築することができる。

次に、第10図および第11図を参照して本発

明に従うマイクロプロセッサの第2実施例について説明する。

前述した本発明の第一の実施例では、リオーグバッファ(RB)15を用いることにより、本発明の目的を達成したが、第2実施例ではリオーグバッファ(RB)15を用いずに本発明の目的を達成するようにしている。

第10図は、第一の実施例の第4図に対応するものであり、第一の実施例と同じ要素には同じ番号をつけてある。第一の実施例と第二の実施例の違いは、次の通りである。

まず、第二実施例は、第一の実施例の構成要素であるリードバッファ(RB)15を削除した構成となっている。そして、第一の実施例のカレントファイル(CF)13とフューチャファイル(FF)14は、第二の実施例では一つの汎用レジスタファイル302になっている。ただしそのエントリ数は16の<X>パートおよび16の<Y>パートの合計32エントリからなる。第二の実施例では、フラグ(FLAG-FLAG1)、エラー情報

(Error-8)、プログラムカウンタ(PC0-8)を一時的に保持する4エントリのステータスファイル301および汎用レジスタファイル(GR)302への書き込み/読み出し番号を生成するGRコントロール回路(GR Control)303を新たに加えている。

以下、第二の実施例の特徴について説明する。

第2の実施例の汎用レジスタ302は先に述べたように、16の<X>パートおよび16の<Y>パートの合計32エントリからなる。<X>パートおよび<Y>パートは、第1の実施例のカレントファイル(CF)13とフューチャファイル(FF)14のように、一つの汎用レジスタR1に対して2本のレジスタ(X1, Y1)を用意している。ただし第一の実施例と異なる点は、<X>パートがカレントファイル、<Y>パートがフューチャファイルと固定的でなく、<X>パートのX1がカレントの値を保持しているレジスタなら、対応する<Y>パートのY1がフューチャの値を保持しているレジスタ、あるいはY1がカレ

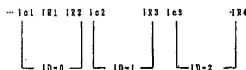
ントの値を保持しているレジスタなら、対応するX1がフューチャの値を保持しているレジスタ、と言う様に、各汎用レジスタR1に対し2本のレジスタ(X1, Y1)がダイナミックにその役割が切り替わることである。

例えば、ある瞬間の<X>のパート、<Y>パートのX1, Y1の役割は次のようになっている。カレントレジスタ値: X0 X1 X2 Y0 X4 Y6 Y7 Y8 X9 X10 X11 X12 X13 Y14 Y15
 フューチャレジスタ値: Y0 Y1 Y2 X3 Y4 Y5 X6 X7 X8 Y9 Y10 Y11 Y12 Y13 X14 X15

第11図は、第10図中のGRコントロール回路(CR Control)303の内部ブロックを示したものである。第11図中には4つのIDレジスタ(304~307)、+1回路308およびGRアドレス生成回路(GR address generator)309から構成される。GRアドレス生成回路309は、バイラインレジスタの汎用レジスタのアクセスに関する情報(310~313)およびIDレジスタの値(314~316)を入力

して汎用レジスタファイル302の読み出し／書き込みアドレス信号(318~321)を出力するブロックである。GRアドレス生成回路309には汎用レジスタファイル302の状態を示す3つのフリップフロップ群(322~324)がある。

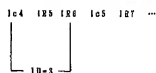
いま、第一の実施例で示した様なパイプライン構成の場合には、プログラムの命令シーケンスと命令実行順序が逆転するのは、「Ic命令に続くIR命令列」であるただしIcはメモリオペランドを持つ命令ないしは実行ステージに数サイクル要する複雑な命令(高機能命令)を示し、IRはメモリオペランドを持たない基本命令を示す。またこの様なパイプライン構成の場合には、IR命令は最大4つのIc命令を飛び越して先に終了する可能性がある。例えばいま命令シーケンスが(先例)



Cなどのステータスもステータスファイル301に一時書き込む。そしてIR命令直前のIc命令が実行ステージを終了するサイクルでIR命令の結果を保持しているX1とY1の役割を切り替える。例えばこの例の場合、Ic1が実行ステージを終了するサイクルでIR1とIR2の結果を保持しているX1とY1の役割を切り替える。

この方法の利点はIc命令に後続するIR命令はハザードが生じないかぎり、いくつでも先行して実行することができ、第一の実施例に見られた様なリオーダーバッファ15のエントリー数による制限が生じないことである。またカレントの値を保持しているレジスタは、GRアドレス生成回路309中のF/F群によってX1またはY1の切り替えを行うため、Ic命令の実行終了時に複数命令のIRの実行が終了している場合にその結果を1サイクルで更新する(すなわちX1とY1の役割を1サイクルで切り替える)ことができる。

次に具体的にどのようにしてX1とY1の役割を切り替え、汎用レジスタファイル302の読み



の場合で、Ic1の命令実行ステージのサイクル数が大きい場合、IR1、IR2はIc1より先に実行が終了し、IR3はIc1、Ic2より先に実行が終了し、IR4はIc1~Ic3より先に実行が終了し、IR5、IR6はIc1~Ic4より先に実行が終了することになる(ただしハザードが生じない場合)。またIR7はIc1の実行が終了するまで実行されない。

この場合問題となるのは、例えばIc1命令実行中に例外が発生した場合、IR1~IR6の実行により更新される汎用レジスタおよびフラグ、PCなどのステータスを元に戻す必要があることである。このためにIR1~IR6の実行結果は、まず汎用レジスタファイル302のカレントの値を保持しているレジスタ(例えばX1)の対のレジスタ(例えばY1)に書き込み、またフラグ、P

出し／書き込みを制御するかについて説明する。

Ic命令とそれに続くIR命令に対して0~3のID番号を割当てる(前述の命令シーケンス列参照)。第11図中のIDレジスタ(304~307)は各パイプラインレジスタ(300~303)中に保持されている命令のID番号を保持している。また汎用レジスタに対して以下の3つのフリップフロップ(F/F)×16のF/F群を設ける。

すなわち、フューチャF/F群322と、有効F/F群323と、ID F/F群324とである。

フューチャF/F群(Future F/F群)322は、16個のF/Fで、汎用レジスタファイル302のX1がカレントの値を保持しているとき対応するフューチャF/Fは1、Y1がカレントの値を保持しているときフューチャF/Fは0となる。

有効F/F群(Valid F/F)323は、16個のF/Fで、フューチャの値(フューチャF

／F1の値が1の時Y1、0の時X1の値)が有効な時1、そうでないとき0となる。

ID F／F群324は、16個で2ビットのF／Fで、フューチャの値が有効なとき、その値を書き込んだ命令のID番号を示す。

GRアドレス生成回路309は、これらF／F群(322～324)の値、パイプラインレジスタの汎用レジスタのアクセス情報(310～313)およびIDレジスタの値(314～316)をもとに汎用レジスタファイル302の読み出し／書き込み番号(318～321)やF／F群の値の更新の制御を次のようにして行う。

1. SEP17で実行されるIR命令の実行に必要なソースオペランドのレジスタR1(SEP34の#src62で指定される)は、対応する有効F／F1=1の時は、フューチャの値(フューチャF／F1の値が1の時Y1、0の時X1の値)、有効F／F1=0の時は、カレントの値(フューチャF／F1の値が1の時X1、0の時Y1の値)とする。

1の時は、フューチャの値(フューチャF／F1の値が1の時Y1、0の時X1の値)、有効F／F1=0の時は、カレントの値(フューチャF／F1の値が1の時X1、0の時Y1の値)とする。ただし有効F／F1=1の時でも対応するID F／FとID4307の時は、ソースオペランドの読み出しは待たされる。

5. IEP(命令実行)ステージでIc命令が終了する時には、そのIc命令と同じID番号を持ち、なおかつ有効F／F1=1の汎用レジスタR1のフューチャF／F1の値を反転し、また有効F／F1を0にリセットする。

6. SEP17で実行されるIR命令の実行結果のレジスタR1(SEP34の#dest63で指定される)のフューチャ(フューチャF／F1の値が1の時Y1、0の時X1)が、このIR命令と異なるID番号やID F／F1の場合には、このIR命令の実行は待たされる。

7. SEP17で実行されるIR命令の実行結果のフラグ(flag)、エラー情報(error)および

2. SEP17で実行されるIR命令の実行結果を格納するディステナーションのレジスタR1(SEP34の#dest63で指定される)は、先行命令が無い(実行が終了している)V78=V87-V96=V101=0)場合には、カレント(フューチャF／F1の値が1の時X1、0の時Y1)、そうでないときはフューチャ(フューチャF／F1の値が1の時Y1、0の時X1)とする。

3. OAG(実行アドレス算出)ステージに必要な汎用レジスタR1(OAGR30のAcode73、Areg74で指定される)は、対応する有効F／F1=1の時は、フューチャの値(フューチャF／F1の値が1の時Y1、0の時X1の値)、有効F／F1=0の時は、カレントの値(フューチャF／F1の値が1の時X1、0の時Y1の値)とする。

4. IEP(命令実行)ステージに必要なソースオペランド(CCUR3のR/M189、#src90で指定される)は、対応する有効F／F1=

FCは、ステータスファイル301のID1レジスタ304の値317で示されるエントリーに一時書き込まれる。そのエントリー番号と同じID番号のIc命令の実行終了時にそれらの値がflag15、error116、PC117にセットされ、更新される。

以上のようにして汎用レジスタファイル302の読み出し／書き込み番号(318～321)やF／F群の更新の制御を行うことにより、比較的簡単なハードウェアで、本発明の目的を達成することができ。

従って、第一実施例の場合、割込みが発生した場合に、フューチャファイル14の値をカレントファイル13の値に直す必要があり、これに最低16サイクルを必要で(汎用レジスタが16本の場合)、これがオーバーヘッドとなり性能低下の原因となっていたが、第二実施例の場合、1つの汎用レジスタ302で行っているため、割込みの発生に対しても、値の移し換えの必要がないものであり、性能低下は起こらない。

また、第一実施例の場合、プログラムシーケンス順で後続する命令が先行する命令を飛び越して実行できる命令数は、リオーダーバッファ15のエントリ数によって制限される。すなわちエントリ数が小さければ性能が低下し、またエントリ数を大きくするとハード量が增加してしまう。

それに対し、第二実施例の場合は、一つの汎用レジスタ302においてXパートとYパートの役割を切り替えて書き込み読み出しを制御しているため、飛び越して実行できる命令数を大きくすることができる。

また、第一実施例の場合、高速分岐の手法として分岐予測を行う場合には、分岐予測が失敗した場合に汎用レジスタの値を元に戻すのに最低16サイクルを必要で、これがオーバーヘッドとなり性能低下の原因となっていたが、第二実施例の場合、汎用レジスタの値を元に戻す必要がないものである。

【発明の効果】

以上説明したように、この発明によれば、第1

の種類の命令と第2の種類の命令を、パイプライン方式によりそれぞれ独立して並列実行処理するようにしたので、パイプラインにおける所定のステージでの稼働率の低下を防止するとともに、パイプラインの乱れを抑制することが可能となる。これにより、性能を大幅に向上させたマイクロプロセッサを提供することができる。

4. 図面の簡単な説明

第1図は、本発明を実施したマイクロプロセッサの内部全体構成を示すブロック図、

第2図は、第1図に示したマイクロプロセッサにおける要部ブロック図、

第3図は、第2図に示すブロック図の各ブロックの内部をさらに詳細に示したブロック図、

第4図は、第3図におけるIL、RB、CF、FFの詳細図、

第5図は、第4図における制御回路の詳細図、

第6図は、第5図に示す状態制御回路の詳細図、

第7図および第8図は、本発明の実施例におけるパイプライン処理動作のタイミング図、

第9図は、本発明の実施例を適用したMPUと周辺LSIから成るシステム構成図、

第10図は、本発明に従うマイクロプロセッサの第二実施例の要部構成図、

第11図は、第10図におけるGRコントロール回路の詳細図、

第12図および第13図は、従来例におけるパイプライン処理動作のタイミング図である。

1…命令フェッチユニット(IFU)

2…デコードユニット(DCU)

3…命令発行ユニット(IU)

4…命令実行ユニット(EU)

5…メモリ管理ユニット(MMU)

6…キャッシュ制御ユニット(CCU)

7…入出力部(I/O)

10…命令デコーダ

11…デコード済命令ループバッファ(DIL)

12…命令発行制御回路(IL)

13…カレントファイル(CF)

14…フューチャファイル(FF)

15…リオーダーバッファ(RB)

17…基本命令実行部(SEP)

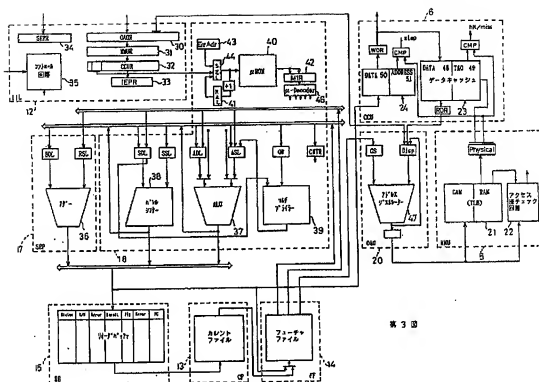
18…高機能命令実行部(IEP)

20…実行アドレス生成部(OAG)

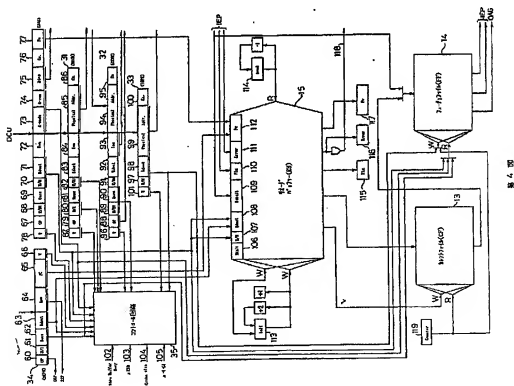
21…アドレス変換バッファ(TLB)

23…データキャッシュメモリ

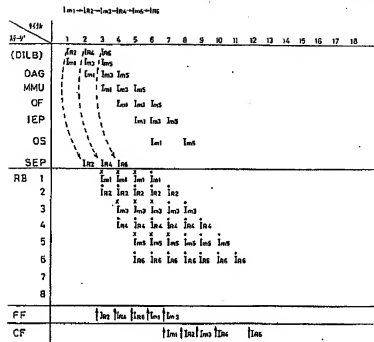
代理人 弁護士 三好 秀 和



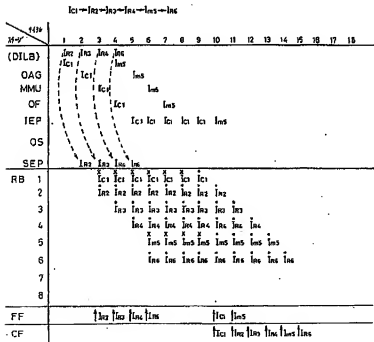
第 3 図



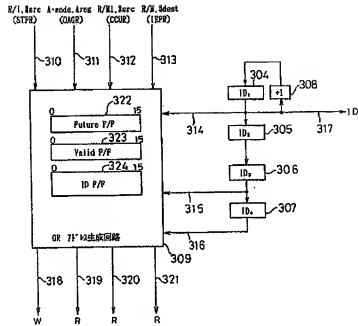
第 4 図



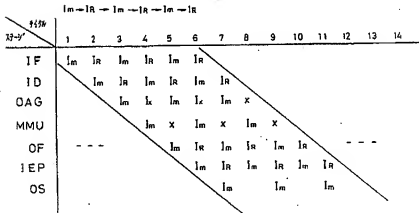
第 7 図



第 8 図



第 11 図



第 12 図

	lc → la → lb → lc → ld													
項目	1	2	3	4	5	6	7	8	9	10	11	12	13	14
IF	lc	la	lb	lc	ld									
ID		lc	la	lb	lc	ld								
OAG			lc	x	x	x	ld							
MMU				lc	x	x	x	ld						
OF					lc	x	x	x	la	lb	lc	ld		
IEP						lc	lc	lc	lc	la	lb	lc	ld	
OS											lc			ld

第13図